

Docket No.: P-198

DJ.
#4 8-3-01
Priority Papers
PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

Jae-Han PARK

Serial No.: New U.S. Patent Application

Filed: March 9, 2001

For: INAP PROCESSING METHOD FOR COMMUNICATION BETWEEN SSP
AND TCAP



TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

Assistant Commissioner of Patents
Washington, D. C. 20231

Sir:

At the time the above application was filed, priority was claimed based on the
following application:

Korean Patent Application No. 11852/2000, filed March 9, 2000

A copy of each priority application listed above is enclosed.

Respectfully submitted,
FLESHNER & KIM, LLP

Daniel Y.J. Kim
Registration No. 36,186

P. O. Box 221200
Chantilly, Virginia 20153-1200
703 502-9440
Date: March 9, 2001
DYK/cba

A large, stylized handwritten signature in black ink, appearing to be "Daniel Y.J. Kim", written over the printed name and registration number.



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

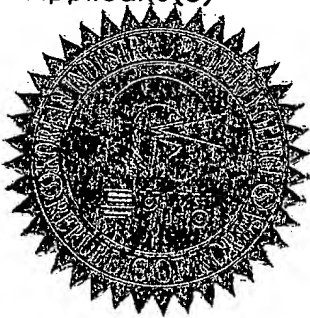
This is to certify that the following application annexed hereto is a true copy from the records of the Korean Industrial Property Office.

JC713 U.S. PTO
09/801836
03/09/01

출원 번호 : 특허출원 2000년 제 11852 호
Application Number

출원 년 월 일 : 2000년 03월 09일
Date of Application

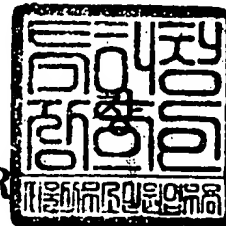
출원인 : 엘지정보통신주식회사
Applicant(s)



2001 년 02 월 27 일

특 허 청

COMMISSIONER



CERTIFIED COPY OF
PRIORITY DOCUMENT

【서류명】 특허출원서
【권리구분】 특허
【수신처】 특허청장
【제출일자】 2000.03.09
【발명의 명칭】 에스에스피에서 타씨에이피와 통신하기 위한 아이엔에이피 처리 방법
【발명의 영문명칭】 INAP Processing Method For TCAP Communication In SSP
【출원인】
【명칭】 엘지정보통신 주식회사
【출원인코드】 1-1998-000286-1
【대리인】
【성명】 김영철
【대리인코드】 9-1998-000040-3
【포괄위임등록번호】 1999-010680-1
【발명자】
【성명의 국문표기】 박재한
【성명의 영문표기】 PARK, Jae Han
【주민등록번호】 700317-1109121
【우편번호】 431-058
【주소】 경기도 안양시 동안구 달안동 셋별한양아파트 612동 1307호
【국적】 KR
【취지】 특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인 김영철 (인)
【수수료】
【기본출원료】 20 면 29,000 원
【가산출원료】 22 면 22,000 원
【우선권주장료】 0 건 0 원
【심사청구료】 0 항 0 원
【합계】 51,000 원
【첨부서류】 1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

본 발명은 SSP(Service Switching Point)에서 SCP(Service Control Point)와의 통신시 INAP(Intelligent Network Application Protocol) 오퍼레이션을 교환하기 위해 TCAP(Transaction Capabilities Application Part)와 상호 작용하면서 INAP 메시지를 처리할 수 있도록 한 SSP에서 TCAP와 통신하기 위한 INAP 처리 방법에 관한 것이다.

종래에는 INAP 오퍼레이션을 교환하는 INAP와 TCAP에 대한 일반적인 권고만을 제시할 뿐, 해당 INAP 오퍼레이션을 교환하기 위한 INAP와 TCAP의 실제 구현에 대한 권고가 제시되어 있지 않았다.

본 발명은 SSP에서 INAP을 이용하는 각종 어플리케이션들이 INAP 메시지를 이용하여 TCAP와 자유롭게 통신할 수 있게 되며, 해당 INAP 처리를 위한 소프트웨어 블록의 객체지향적인 설계, 구현을 통해 재사용성을 증대시킴으로써, 이를 이용하는 어플리케이션의 변경없이 INAP 메시지만을 추가, 변경, 삭제하여 새로운 시스템에 적용할 수 있게 됨과 동시에 유지 보수가 간편해진다.

【대표도】

도 8

【명세서】**【발명의 명칭】**

에스에스피에서 티씨에이피와 통신하기 위한 아이엔에이피 처리 방법{INAP Processing Method For TCAP Communication In SSP}

【도면의 간단한 설명】

도 1은 일반적인 지능망에서 SSP와 SCP 사이의 INAP 오퍼레이션 교환 구조를 도시한 도면.

도 2는 본 발명에 따른 SSP에서 TCAP와 통신하기 위한 INAP 처리 장치의 소프트웨어적인 구조를 개략적으로 도시한 도면.

도 3은 도 2에 있어, TCAP 다이얼로그 블록을 상세하게 도시한 도면.

도 4는 도 2에 있어, INAP 메시지 블록을 상세하게 도시한 도면.

도 5는 도 2에 있어, TCAP 이벤트 블록을 상세하게 도시한 도면.

도 6은 도 5에 있어, TC 프리미티브 객체를 상세하게 도시한 도면.

도 7은 도 2에 있어, TCAP 다이얼로그 관리 블록을 상세하게 도시한 도면.

도 8은 본 발명에 따른 INAP 처리 장치에 의한 INAP 메시지의 송신 절차를 도시한 도면.

도 9는 도 8에 있어, 송신 TCAP 이벤트의 실행 절차를 도시한 도면.

도 10은 도 9에 있어, TCAP 메시지의 송신 절차를 도시한 도면.

도 11은 본 발명에 따른 INAP 처리 장치에 의한 INAP 메시지의 수신 절차를 도시한 도면.

도 12는 도 11에 있어, TC 컴포넌트에 대한 처리 절차를 도시한 도면.

도 13은 도 12에 있어, TCAP 메시지의 TC 프리미티브 실행 절차를 도시한 도면.

도 14는 도 12에 있어, TCAP 메시지의 TC 컴포넌트 실행 절차를 도시한 도면.

도 15는 도 13에 있어, SCP와의 다이얼로그를 종료시키는 절차를 도시한 도면.

* 도면의 주요 부분에 대한 부호의 설명 *

20 : TCAP 인터페이스 블록

30 : TCAP 다이얼로그 블록

31 : TCAP 다이얼로그 객체

32 : TCAP 메시비 객체

33 : SCP 관계 객체

34 : 상위 TCAP 다이얼로그 상태 객체

40 : INAP 메시지 블록

41 : TC 컴포넌트 객체

42 : TC 취소 객체

43 : TC 에러 객체

44 : TC 거절 객체

45 : TC 결과 객체

46 : TC 인보크 객체

50 : TCAP 이벤트 블록

51 : TCAP 다이얼로그 이벤트 객체

52 : TC 프리미티브 객체

53 : 송신 TCAP 객체

54 : 송신 종료 TCAP 객체

60 : TCAP 다이얼로그 관리 블록

61 : TCAP 다이얼로그 관리 객체

62 : 맵 관리 객체

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

- <28> 본 발명은 SSP(Service Switching Point)에서 TCAP(Transaction Capabilities Application Part)와 통신하기 위한 INAP(Intelligent Network Application Protocol) 처리 방법에 관한 것으로, 특히 SSP에서 SCP(Service Control Point)와의 통신시 INAP 오퍼레이션을 교환하기 위해 TCAP와 상호 작용하면서 INAP 메시지를 처리할 수 있도록 한 SSP에서 TCAP와 통신하기 위한 INAP 처리 방법에 관한 것이다.
- <29> 일반적으로, 신속한 서비스의 생성과 효율적인 서비스 제어를 목적으로 네트워크의 기능이 다수개의 시스템으로 분산되는 구조를 갖는 지능망에서는 각 시스템간의 정보 교환을 위해 지능망 응용 프로토콜인 INAP을 권고하고 있는데, 해당 INAP은 물리 실체들간의 상호 작용을 지원하기 위한 응용 계층의 프로토콜이며, INAP의 하부 프로토콜로서는 No.7 공통선 신호 방식의 프로토콜이 주로 사용되고 있다.
- <30> 즉, 해당 INAP은 No.7 공통선 신호 방식을 이용하여 SCP-SSP, SCP-IP(Intelligent Peripheral)와 같은 물리 실체들 사이의 상호 작용을 지원하기 위한 TCAP 상위의 응용 계층 프로토콜로서, 해당 SSP와 SCP는 첨부된 도면 도 1에 도시된 바와 같이 No.7 공통선 신호망을 통해 TCAP 메시지로 INAP 오퍼레이션을 교환할 수 있었다.
- <31> 그런데, 이러한 INAP을 권고하고 있는 ITU-T(International Telecommunication Union-Telecommunication standardization section)에서는 해당 INAP 오퍼레이션을 교환

하는 INAP와 TCAP에 대한 일반적인 권고만을 제시할 뿐, 해당 INAP 오퍼레이션을 교환하기 위한 INAP와 TCAP의 실제 구현에 대한 권고는 제시하지 않았다.

【발명이 이루고자 하는 기술적 과제】

- <32> 본 발명은 전술한 바와 같은 문제점을 해결하기 위한 것으로 그 목적은, SSP에서 INAP을 이용하는 각종 어플리케이션들이 INAP 메시지를 이용하여 TCAP와 자유롭게 통신할 수 있는 INAP 처리 방법을 제공하는데 있다.
- <33> 본 발명의 다른 목적은, SSP에서 INAP 처리를 위한 소프트웨어 블록의 객체지향적인 설계, 구현을 통해 재사용성을 증대시킴으로써, 이를 이용하는 어플리케이션의 변경 없이 INAP 메시지만을 추가, 변경, 삭제하여 새로운 시스템에 적용할 수 있도록 함과 동시에 유지 보수의 편리성을 제공하는데 있다.

【발명의 구성 및 작용】

- <34> 상술한 바와 같은 목적을 해결하기 위한 본 발명의 특징은, INAP 팩터리 객체를 통해 INAP 초기 감지점과 관련된 INAP 객체를 생성하는 과정과; 상기 INAP 객체를 파라미터로 송신 컴포넌트 함수를 호출하여 인보크 ID와 다이얼로그 ID를 TCAP 다이얼로그 객체로부터 얻어 상기 INAP 메시지에 셋팅하는 과정과; 컴포넌트 추가 명령을 내려 TCAP 메시지 객체에 상기 INAP 객체를 추가하는 과정과; 현재의 TCAP 다이얼로그 상태에 따라서 다른 송신 TCAP 이벤트를 생성 및 실행시켜 상기 INAP 메시지를 송신한 후, 삭제하는 과정을 포함하는 에스에스피에서 티씨에이피와 통신하기 위한 아이엔에이피 처리 방

법을 제공하는데 있다.

<35> 본 발명의 다른 특징은, TCAP으로부터 TC 프리미티브가 수신되는 경우 수신된 TC 프리미티브를 처리하기 위한 TC 시작 객체를 생성하는 과정과; 상기 TC 시작 객체를 실행시켜 수신된 TC 프리미티브에 포함된 다이얼로그 ID를 이용하여 해당되는 SCP 관계 객체를 찾는 과정과; 상기 SCP 관계 객체로 TC 시작 추가를 명령하여 TCAP 메시지에 TC 시작 객체를 추가시키는 과정과; TC 컴포넌트없이 TC 프리미티브만 수신된 경우 상기 TC 시작 객체가 추가된 TC 프리미티브를 실행시키는 과정과; 해당되는 호 처리를 수행한 후, 상기 SCP 관계 객체를 삭제하여 SCP와의 다이얼로그를 종료시키는 과정을 포함하는 에스에스피에서 티씨에이피와 통신하기 위한 아이엔에이피 처리 방법을 제공하는데 있다.

<36> 이하, 본 발명에 따른 실시예를 첨부한 도면을 참조하여 상세하게 설명하면 다음과 같다.

<37> 본 발명에 따른 지능망에서 SSP는 SCP와의 통신시 INAP, TCAP, SCCP(Signaling Connection Control Part), MTP(Message Transfer Part) 등의 다단계 프로토콜을 이용하게 되는데, 이때, INAP 오퍼레이션 교환을 위해 TCAP와 상호 작용하면서 INAP 메시지를 처리하는 INAP 처리 장치의 소프트웨어적인 구성은 첨부

한 도면 도 2와 같이, TCAP와 정합하여 INAP 메시지를 송/수신하는 TCAP 인터페이스 블록(20)과, SSP와 SCP 사이의 통신 채널인 다이얼로그를 각각의 다이얼로그 상태별로 관리하는 TCAP 다이얼로그 블록(30)과, 다수개의 객체(object)들이 상속받으면서 계층 구조로 연결되어 TCAP와 송/수신되는 INAP 메시지를 처리하는 INAP 메시지 블록(40)과, TCAP와 송/수신되는 INAP 메시지와 함께 생성되는 TCAP 다이얼로그 이벤트를 관리하는 TCAP 이벤트 블록(50) 및 TCAP와 INAP이 다이얼로그 단위로 상호 통신시 각각의 다이얼로그를 구분할 다이얼로그 ID(IDentifier)를 생성 및 관리하는 TCAP 다이얼로그 관리 블록(60)을 구비하여 이루어진다.

<38> 해당 TCAP 인터페이스 블록(20)은 하나의 TCAP 인터페이스 객체로 구성되며, 해당 TCAP 인터페이스 객체는 TCAP와 직접적으로 인터페이스하는 부분으로 INAP 메시지 블록(40) 및 TCAP 이벤트 블록(50)에 의해 사용된다.

<39> 그리고, 해당 TCAP 다이얼로그 블록(30)과, INAP 메시지 블록(40)과, TCAP 이벤트 블록(50) 및 TCAP 다이얼로그 관리 블록(60)은 다수개의 객체들을 더 구비하며, 각각의 소프트웨어 블록에 구성된 객체들을 통해 상호 연동하여, 네트워크 상에서 두 개의 노드가 INAP, TCAP을 사용하여 통신하는 경우에 필요한 논리적인 통신 경로인 다이얼로그 관련 INAP 메시지 처리를 수행하게 되는데, 이를 첨부한 도면을 참조하여 보다 상세히 설명하면 다음과 같다.

<40> 첫째로, 해당 TCAP 다이얼로그 블록(30)은 첨부한 도면 도 3에 도시한 바와 같이 TCAP 다이얼로그 객체(31)와, TCAP 메시지 객체(32)와, SCP 관계 객체(33) 및 상위 TCAP 다이얼로그 상태 객체(34)를 더 구비하되, 각각의 다이얼로그 상태(휴지

상태, 초기 상태, 전이중 상태, 종료 상태)를 나타내는 객체들을 포함하며, 해당 다이얼로그에는 자신이 송신한 다이얼로그에 대한 응답이 필요없는 단방향 메시지를 송신할 때 사용하는 비구조화된 다이얼로그와, 자신이 송신한 다이얼로그에 대한 응답이 필요한 양방향의 다이얼로그를 형성하여 각각의 다이얼로그 ID로 구분되는 다수개의 다이얼로그에 대한 시작, 수행, 종료의 과정을 동시에 처리할 수 있는 구조화된 다이얼로그가 있다.

<41> 여기서, 해당 TCAP 다이얼로그 객체(31)는 상위 TCAP 다이얼로그 상태 객체(34)를 통해 현재 다이얼로그에 대한 상태를 가지고 있으며, SCP 관계 객체(33)의 베이스 객체가 된다.

<42> 해당 TCAP 메시지 객체(32)는 INAP 메시지 블록(40)의 TC 컴포넌트 객체와 연동하여 TCAP와 송/수신할 INAP 메시지를 저장 및 처리하는데, 여기서, TC 컴포넌트 객체는 수신된 INAP 메시지를 처리할 TC 동작(TC 취소, TC 에러, TC 거절, TC 결과, TC 인보크) 객체의 베이스가 되는 객체로서, 각 동작 객체들이 공통적으로 사용하는 데이터 및 메소드(method) 즉, 송/수신되는 INAP 메시지를 실행시키는 다수의 메소드를 포함한다.

<43> 해당 SCP 관계 객체(33)는 SCP와 통신을 시작할 때마다 TCAP 다이얼로그 객체(31)로부터 속성을 상속받아 새로 생성되어 해당 SCP와의 관계를 나타내며, 해당 SCP에 대한 통신 경로로서 실제 물리적(physical)으로 존재하는 SCP에 대한 네트워크 상의 어드레스 값과 메시지 처리에 관련된 함수들을 포함한다.

<44> 해당 상위 TCAP 다이얼로그 상태 객체(34)는 TCAP 다이얼로그 상태를 나타내

는 휴지 상태 객체(34-1), 초기 상태 객체(34-2), 전이중 상태 객체(34-3), 종료 상태 객체(34-4)들을 상속하고 있는 상위 클래스로서, 각각의 상태 객체에 대한 공통적인 속성을 포함하는데, 해당 휴지 상태 객체(34-1)는 다이얼로그를 시작하는 TC 시작을 나타내는 TC 프리미티브를 송신할 수 있는 상태 객체로서, TCAP 다이얼로그가 최초로 생성될 때 천이되며, 해당 초기 상태 객체(34-2)는 휴지 상태 객체(34-1)에서 TC 시작을 송신할 때 천이된다. 해당 전이중 상태 객체(34-3)는 TC 시작을 송신한 후 TC 수행을 수신하거나, 최초로 TC 시작을 최초로 수신할 때 천이되며, 해당 종료 상태 객체(34-3)는 TC 종료나 TC 중지와 같이 다이얼로그 종료를 요구하는 TC 프리미티브를 수신하거나, 진행되는 호가 종료될 때 천이된다.

<45> 둘째로, 해당 INAP 메시지 블록(40)은 첨부한 도면 도 4와 같이, TCAP와 송/수신하는 INAP 메시지인 TC 컴포넌트 객체(41)와, 해당 TC 컴포넌트 객체(41)에 의해 상속되는 다수개의 TC 동작 객체(42~46)들이 계층 구조로 연결되어 있으며, 최상위 계층에 TC 컴포넌트 객체(41)가 위치하고, 해당 TC 컴포넌트 객체(41)는 수신된 INAP 메시지를 처리할 TC 동작을 나타내는 TC 동작 객체인 TC 취소 객체(42)와, TC 에러 객체(43)와, TC 거절 객체(44)와, TC 결과 객체(45) 및 TC 인보크 객체(46)를 상속한다.

<46> 해당 TC 취소 객체(42)는 이전에 송신한 INAP 메시지를 취소할 때 사용하고, 해당 TC 에러 객체(43)는 INAP 메시지 처리를 실패했을 때 사용하며, 해당 TC 거절 객체(44)는 INAP 메시지에 대해 문제가 발생하여 이를 거절할 때 사용한다. 해당 TC 결과 객체(45)는 INAP 메시지에 대한 정상적인 처리 결과를 송/수신할 때 사용하며, 해당 TC 인보크 객체(46)는 INAP 메시지를 통해 소정의 동작을 수행시킬 때 사용한다.

<47> 나아가, 해당 TC 결과 객체(45)는 INAP 결과 객체(45-1)가 상속받되, 템플릿으로

선언되어 사용되는 개별적인 INAP 결과에 따라 서로 다른 INAP 결과 객체를 생성하고, 해당 TC 인보크 객체(46)는 INAP 객체(46-1)가 상속받되, 템플릿으로 선언되어 사용되는 개별적인 INAP 동작 타입에 따라 서로 다른 INAP 객체를 생성한다. 그리고, 해당 INAP 객체(46-1)는 INAP 팩터리(factory) 객체(46-2)를 사용하며, 해당 INAP 팩터리 객체(46-2)는 템플릿 종류에 따라 서로 다른 종류의 INAP 객체를 생성한다.

<48> 예를 들어, TC 결과 객체(45)로 송신될 수 있는 결과로 수신 정보 결과와 요구된 현재 상태 보고 결과가 있는 경우 각각의 결과에 대해 템플릿으로 선언된 INAP 결과 객체(45-1)가 생성되고, 이후에 해당 INAP 결과 객체(45-1)의 베이스가 되는 TC 결과 객체(45)가 생성되며, 다시 해당 TC 결과 객체(45)의 베이스가 되는 TC 컴포넌트 객체(41)가 생성됨에 따라 이를 이용하여 INAP 메시지를 송신할 있어 서로 다른 종류의 TC 결과 객체(45)가 생성될 수 있다.

<49> 그리고, INAP 객체(46-1)의 템플릿으로 초기 감지점 INAP 동작 타입을 사용하는 경우 이에 대해 템플릿으로 선언된 INAP 객체(46-1)가 INAP 팩터리 객체(46-2)에 의해 생성되고, 이후에 해당 INAP 객체(46-1)의 베이스가 되는 TC 인보크 객체(46)가 생성되며, 다시 해당 TC 인보크 객체(46)의 베이스가 되는 TC 컴포넌트 객체(41)가 생성됨에 따라 이를 이용하여 INAP 메시지를 생성함으로써, 해당 TC 인보크 객체(46)로 대표되는 서로 다른 종류의 INAP 객체(46-1)가 생성될 수 있다.

<50> 셋째로, 해당 TCAP 이벤트 블록(50)은 첨부한 도면 도 5와 같이 TCAP 다이얼로그 이벤트 객체(51)와, 해당 TCAP 다이얼로그 이벤트 객체(51)에 의해 상속되는 TC 프리미티브 객체(52)와 송신 TCAP 객체(53) 및 송신 종료 TCAP 객체(54)를 더 구비하되, 해당 TCAP 다이얼로그 이벤트 객체(52)는 각종 TCAP 다이얼로그 관련 이벤트의 베이스가

되며, 해당 TCAP 다이얼로그 관련 이벤트의 공통된 속성을 갖는다.

<51> 이때, 해당 TC 프리미티브 객체(52)는 TCAP으로부터 INAP 메시지를 수신하는 경우에 발생하는 TC 프리미티브 이벤트를 처리하되, TCAP와의 다이얼로그에 대한 성격 즉, TC 시작, TC 수행, TC 종료, TC 중지를 나타내는 객체를 생성하며, 해당 TC 프리미티브는 TC 컴포넌트와 함께 TCAP으로 송/수신되는 INAP 메시지로서, 각각의 다이얼로그 성격에 대한 공통 속성을 나타낸다.

<52> 여기서, 해당 다이얼로그 성격을 나타내는 각 객체를 첨부한 도면 도 6을 참조하여 설명하면, 해당 TC 시작 객체(52-1)는 TCAP와의 다이얼로그를 시작할 때 생성되며, 해당 TC 수행 객체(52-2)는 TC 시작 객체(52-1)에 대한 응답을 송신할 때 생성된다. 해당 TC 종료 객체(52-3)는 다이얼로그를 종료할 때 생성되며, TC 중지 객체(52-4)는 메시지 또는 네트워크의 이상 발생 등으로 다이얼로그를 강제로 종료할 때 생성된다.

<53> 해당 송신 TCAP 객체(53)는 INAP 메시지(TC 프리미티브와 TC 컴포넌트)를 송신할 때 생성되는 이벤트 객체로서, 현재의 다이얼로그 상태에 따라 다이얼로그의 상태를 유지 또는 변경한다.

<54> 해당 송신 종료 TCAP 객체(54)는 INAP 메시지를 송신한 후 현재의 다이얼로그를 종료하고자 할 때 생성하는 것으로, 이때 송신하는 INAP 메시지의 TC 프리미티브는 TC 종료 객체(52-3)가 된다.

<55> 마지막으로, 해당 TCAP 다이얼로그 관리 블록(60)은 첨부한 도면 도 7과 같이, INAP와 TCAP이 상호 통신시 사용하는 각각의 다이얼로그 ID 및 해당 다이얼로그 ID에 해당하는 SCP 관계 객체를 생성하는 TCAP 다이얼로그 관리 객체(61)와, 생성한 각각의 다

이얼로그 ID에 해당되는 SCP 관계 객체(33)에 대한 정보를 저장 및 관리하는 맵 관리 객체(62)를 더 구비한다.

<56> 상술한 바와 같이 구성된 본 발명에 따른 SSP에서 TCAP와 통신하기 위한 INAP 처리 장치의 INAP 메시지 처리 절차를 첨부한 도면을 참조하여 상세하게 설명하면 다음과 같다.

<57> 첫째로, INAP 처리 장치에 의한 INAP 메시지의 송신 처리 절차는 첨부한 도면 도 8과 같이 진행된다.

<58> 먼저, INAP을 사용하는 어플리케이션인 호 처리 객체가 INAP 팩터리 객체(46-2)를 통해 INAP 초기 감지점(INAP_InitialDP)과 관련된 INAP 객체(46-1)를 생성하게 되는데, 이를 위해 해당 INAP 팩터리 객체(46-2)는 자신이 가지고 있는 템플릿 종류에 따른 다수개의 INAP 객체 관련 함수 중에서 초기 감지점 생성 함수(createInitialDP())를 호출하여(스텝 S1), C++ 프로그래밍에 의한 'NEW' 연산자를 이용한 INAP 객체 생성 절차를 통해 INAP 초기 감지점과 관련된 INAP 객체(46-1)를 생성하게 된다(스텝 S2).

<59> 이후, 해당 호 처리 객체는 생성한 INAP 객체(46-1)를 TCAP으로 전달하기 위해 SCP 관계 객체(33)와 관련된 송신 컴포넌트 함수(sendComponent())를 호출한 후(스텝 S3), 해당 INAP 메시지를 송신하기 위해 필요한 인보크 ID와 다이얼로그 ID를 TCAP 다이얼로그 객체(31)로부터 얻어 해당 INAP 메시지에 셋팅하게 된다(스텝 S4, S5).

<60> 이때, 해당 INAP 메시지는 TCAP 다이얼로그 객체(31)를 통해 송신되므로, 해당 INAP 메시지 블록(40)의 INAP 객체(46-1)는 자신을 파라미터로 하여 TCAP 다이얼로그 객체(31)의 송신 컴포넌트 함수를 호출하게 된다(스텝 S6).

- <61> 이에, 해당 TCAP 다이얼로그 객체(31)는 자신이 가지고 있는 TCAP 메시지 객체(32)에 컴포넌트 추가(addComponent) 명령을 내려, 해당 TCAP 메시지 객체(32)에 INAP 객체(46-1)를 추가하게 된다(스텝 S7).
- <62> 그리고, 해당 TCAP 다이얼로그 객체(31)는 TCAP 메시지 객체(32)에 추가된 INAP 객체(46-1)를 송신하기 위해 송신 컴포넌트 함수를 호출하여(스텝 S8), TCAP 다이얼로그 이벤트인 송신 TCAP 이벤트를 생성하고(스텝 S9), 해당 송신 TCAP 이벤트를 실행시켜 INAP 메시지를 송신한 후(스텝 S10), 이를 삭제하게 된다(스텝 S11).
- <63> 한편, 스텝 S10에서, 해당 송신 TCAP 이벤트의 실행 절차를 첨부한 도면 도 9를 참조하여 보다 상세히 설명하면 다음과 같다.
- <64> 해당 TCAP 다이얼로그 이벤트 객체(51)는 현재의 TCAP 다이얼로그 상태에 따라서 서로 다른 이벤트 실행을 수행하게 되는데, 해당 TCAP 다이얼로그 이벤트인 송신 TCAP 이벤트는 상위 TCAP 다이얼로그 상태 객체(34)에 의해 휴지 상태에서 생성되며(스텝 S21), 이에 대응하는 송신 TCAP 함수(doSendTCAP())가 상위 TCAP 다이얼로그 상태 객체(34)에 의해 호출된다(스텝 S22).
- <65> 이때, 해당 INAP 객체(46-1)는 TC 컴포넌트 객체(41)로부터 상속받게 되며, 해당 TC 컴포넌트는 TC 프리미티브와 함께 송신해야 하므로, TC 프리미티브 객체(52)를 통해 TC 시작 객체(52-1)를 생성하게 된다(스텝 S23). 즉, 송신 TCAP 이벤트는 휴지 상태에서 TC 시작을 나타내는 TC 프리미티브 객체(52)를 생성하게 된다.
- <66> 이후, 해당 상위 TCAP 다이얼로그 상태 객체(34)는 SCP 관계 객체(33)로부터 TC 시작에 필요한 정보(SCP의 네트워크 어드레스 등)를 얻어 TC 프리미티브에 셋팅한 후(스텝

S24), 자신이 가지고 있는 TC 시작 객체(52-1)에 프리미티브 추가 명령을 내려, 이미 TC 컴포넌트가 추가된 TCAP 메시지에 TC 프리미티브를 추가하게 된다(스텝 S25).

<67> 그리고, 해당 상위 TCAP 다이얼로그 상태 객체(34)는 TC 컴포넌트 및 TC 프리미티브가 추가된 TCAP 메시지를 송신하기 위해 TCAP 다이얼로그 객체(31)로부터 TCAP 메시지 송신 함수(sendTCAPmsg())를 호출하게 되고(스텝 S26), 이에, 해당 TCAP 다이얼로그 객체(31)는 휴지 상태에서 TCAP 메시지 즉, TC 컴포넌트 및 TC 프리미티브를 송신하게 된다(스텝 S27).

<68> 이에 따라, 해당 상위 TCAP 다이얼로그 상태 객체(34)는 TCAP 다이얼로그 객체(31)로 초기 상태 천이 명령을 내려(스텝 S28), 현재 휴지 상태인 TCAP 다이얼로그 객체(31)를 초기 상태로 천이하게 된다(스텝 S29).

<69> 나아가, 상술한 스텝 S27에서 해당 TCAP 메시지의 송신 절차는 TC 컴포넌트 및 TC 프리미티브의 순서로 수행되는데, 이를 첨부한 도면 도 10을 참조하여 보다 상세하게 설명하면 다음과 같다.

<70> 해당 TCAP 메시지 객체(32)가 TC 컴포넌트를 TCAP 및 그 하부 네트워크 프로토콜에서 사용할 수 있는 포맷으로 가공하여 TCAP으로 송신하기 위해 TC 인보크 객체(46)로부터 인코딩 함수를 호출하게 되면(스텝 S31), 해당 TC 인보크 객체(46)는 실제 TCAP으로 송신할 인보크 요구 메시지를 생성한 후(스텝 S32), 해당 인보크 요구 메시지의 데이터 영역에 가공한 TC 컴포넌트를 실는다(스텝 S33).

<71> 이후, 해당 TC 인보크 객체(46)는 TCAP와의 통신을 지원하는 라이브러리 함수를 호출하여(스텝 S34), 해당 TCAP 메시지의 TC 컴포넌트를 TCAP으로 송신하게 된다.

- <72> 한편, 해당 TC 컴포넌트를 모두 TCAP으로 송신한 후, 해당 TCAP 메시지 객체(32)는 TC 프리미티브를 송신하기 위해 TC 프리미티브 객체(52)로부터 인코딩 함수를 호출하게 되고(스텝 S35), 이에, 해당 TC 프리미티브 객체(52)는 TC 시작 상태로 천이하여 TC 프리미티브를 TCAP 및 그 하부 네트워크 프로토콜에서 사용할 수 있는 포맷으로 가공하게 된다.
- <73> 이후, 해당 TC 프리미티브 객체(52)는 실제 TCAP으로 송신할 시작 요구 메시지를 생성하여(스텝 S36), 해당 시작 요구 메시지의 데이터 영역에 가공한 TC 프리미티브를 실은 후, 해당 TCAP와의 통신을 지원하는 라이브러리 함수를 호출하여, 해당 TCAP 메시지의 TC 프리미티브를 TCAP으로 송신하게 된다(스텝 S37).
- <74> 둘째로, 본 발명에 따른 INAP 처리 장치에 의한 INAP 메시지의 수신 처리 절차는 첨부한 도면 도 11과 같이 진행되는데, 해당 TCAP으로부터 TC 컴포넌트와 TC 프리미티브 및 기타 INAP이 사용하는 하부 네트워크의 상태 관련 메시지 등과 같은 INAP 메시지는 TCAP 인터페이스 객체를 통해 수신되며, 수신된 INAP 메시지는 이를 사용하는 각각의 어플리케이션에 분배된다.
- <75> 이때, 해당 INAP 메시지의 수신 처리는 먼저 TC 프리미티브를 처리하면서 TCAP 다 이얼로그 객체의 상태를 천이 또는 유지하고, 다음에 TC 컴포넌트를 처리하게 된다.
- <76> 한편, 해당 TCAP으로부터 TCAP 인터페이스 객체로 TC 시작을 나타내는 TC 프리미티브가 수신되면, 해당 TCAP 인터페이스 객체를 관리하는 TCAP 인터페이스 관리 객체(도면에 도시되어 있지 않음)는 시작 관련 메시지 처리 함수(handleBeginIND(msg))를 호출하여(스텝 S41), 수신된 TC 시작을 나타내는 TC 프리미티브를 처리하기 위해 TC 프리미티브 객체(52)를 통해 새로운 TC 시작 객체(52-1)를 생성하되(스텝 S42), 해당 TC 시작 객

체(52-1)에 수신된 TC 프리미티브를 디코딩하여(스텝 S43), 이에 포함된 TC 시작 객체와 동일한 TC 시작 객체(52-1)를 생성하게 된다.

<77> 그리고, 해당 TCAP 인터페이스 관리 객체는 수행 함수(executeIND())를 호출하여(스텝 S44), 생성한 TC 시작 객체(52-1)를 실행시키게 되며, 해당 TC 시작 객체(52-1)는 수신된 TC 프리미티브에 포함된 다이얼로그 ID를 이용하여 TCAP 다이얼로그 관리 객체(61)를 통해 해당되는 SCP 관계 객체(33)를 찾게 되는데(스텝 S45), 이때, 해당 다이얼로그 ID를 찾지 못한 경우에는 TCAP 다이얼로그 관리 객체(61)를 통해 새로운 SCP 관계 객체(33)를 생성하게 된다(스텝 S46).

<78> 즉, 해당 TC 시작 객체(52-1)를 상속하는 TC 프리미티브 객체(52)는 다이얼로그 ID를 이용하여 TCAP 다이얼로그 관리 객체(61)로 SCP 관계 객체(33)를 찾도록 명령하거나, 해당 다이얼로그 ID를 찾지 못한 경우에는 TCAP 다이얼로그 관리 객체(61)로 새로운 SCP 관계 객체(33)의 생성을 명령하게 된다.

<79> 이에 따라, 해당 TCAP 다이얼로그 관리 객체(61)는 다이얼로그 ID에 해당되는 SCP 관계 객체(33)를 찾거나 새로운 SCP 관계 객체(33)를 생성함으로써, 해당 SCP 관계 객체(33)를 통해 TCAP 다이얼로그 및 TCAP 다이얼로그 상태를 관리할 수 있게 된다.

<80> 이후, 해당 TC 프리미티브 객체(52)가 SCP 관계 객체(33)로 TC 시작 추가를 명령함에 따라 해당 SCP 관계 객체(33)는 TCAP 메시지에 TC 시작 객체(52-1)를 추가시키게 되며(스텝 S47, S48), 이어서, 해당 TC 프리미티브 객체(52)는 SCP 관계 객체(33)를 통해 추가된 TC 시작 객체(52-1)를 실행시키되(스텝 S49, 50), TC 컴포넌트없이 TC 프리미티브만 수신된 경우에만 TC 시작 객체(52-1)의 실행이 가능하다.

- <81> 만약, 해당 TC 프리미티브와 TC 컴포넌트가 함께 수신된 경우에는 해당 TC 프리미티브인 TC 시작 객체(52-1)의 실행을 보류한 상태에서 해당 TC 컴포넌트를 모두 처리한 후, 보류되었던 TC 시작 객체(52-1)를 실행시키고 다음에 TC 컴포넌트를 차례로 실행시키게 된다.
- <82> 여기서, 해당 TC 컴포넌트에 대한 처리 절차를 첨부한 도면 도 12에 도시한 TC 컴포넌트의 TC 인보크 객체(46)를 처리하는 절차에 따라 설명하면 다음과 같다.
- <83> 해당 TCAP으로부터 TCAP 인터페이스 객체를 통해 TC 컴포넌트의 TC 인보크 객체(46)가 수신되면, 해당 TCAP 인터페이스 객체를 관리하는 TCAP 인터페이스 관리 객체는 인보크 관련 메시지 처리 함수(handleInvokeIND(msg))를 호출하여(스텝 S51), 수신된 TC 인보크 객체(46)를 처리하게 된다.
- <84> 이때, 해당 TC 인보크 객체(46)에는 다양한 종류의 INAP 객체(46-1)가 존재함에 따라 이를 처리하기 위해서는 TC 인보크 객체(46)에 실려있는 오퍼레이션 코드를 분석해야 하며, 해당 오퍼레이션 코드가 INAP 커백션과 관련된 것으로 확인되면, 커백션 처리 함수(handle_Connect(msg))를 호출하여 실행시키게 된다(스텝 S52).
- <85> 이를 위해 해당 TCAP 인터페이스 관리 객체는 C++ 프로그래밍에 의한 'NEW' 연산자를 이용하여 커백션 관련 INAP 객체(46-1)를 새로 생성한 후(스텝 S53), 해당 INAP 객체(46-1)에 수신된 TC 컴포넌트를 디코딩하여(스텝 S54), 송신한 INAP 객체와 동일한 커백션 관련 INAP 객체(46-1)를 생성하게 된다.
- <86> 그리고, 해당 TCAP 인터페이스 관리 객체는 수행 함수를 호출하여 INAP 객체(46-1)를 실행시키게 되며(스텝 S55), 해당 INAP 객체(46-1)는 수신된 TC 컴포넌트에 포함된

다이얼로그 ID를 이용하여 TCAP 다이얼로그 관리 객체(61)를 통해 해당되는 SCP 관계 객체(33)를 찾게 된다(스텝 S56).

<87> 이후, 해당 INAP 객체(46-1)가 SCP 관계 객체(33)로 TC 컴포넌트 추가를 명령함에 따라 해당 SCP 관계 객체(33)는 TCAP 메시지에 TC 컴포넌트를 추가시키게 되며(스텝 S57, S58), 이어서, 해당 INAP 객체(46-1)는 SCP 관계 객체(33)를 통해 추가된 TC 컴포넌트 객체(41)를 실행시키게 되는데(스텝 S59, S60), 이때, 해당 TC 컴포넌트의 실행은 마지막 TC 컴포넌트인 경우에 가능하다.

<88> 한편, 상술한 절차에 따라서 TCAP 메시지 즉, TC 프리미티브의 TC 시작 객체(52-1) 및 TC 컴포넌트의 TC 인보크 객체(46)를 수신받아 처리한 후, 각각을 실행시키게 되는데, 먼저 TC 프리미티브를 실행시킨 후에 TC 컴포넌트를 실행시키게 되며, 이에 대한 실행 절차를 첨부한 도면을 참조하여 상세하게 설명하면 다음과 같다.

<89> 먼저, 해당 TCAP 메시지의 TC 프리미티브 실행 절차는 첨부한 도면 도 13과 같이 진행되는데, 해당 TC 프리미티브는 SCP 관계 객체(33)가 TCAP 메시지 객체(32)로 TC 시작 객체(52-1)의 실행을 명령함으로써 시작된다(스텝 S61).

<90> 해당 TC 시작 객체의 실행을 명령받은 TC 메시지 객체는 TCAP 다이얼로그 객체를 통해 TC 시작을 나타내는 TC 프리미티브 이벤트를 발생시키게 되고(스텝 S62), 이에, TC 프리미티브 객체(52)는 상위 TCAP 다이얼로그 상태 객체(34)로부터 현재의 TCAP 다이얼로그 상태 즉, TCAP 다이얼로그 객체(31)를 통해 휴지 상태를 리턴받은 후(스텝 S63), 해당 휴지 상태에서 TC 시작 이벤트에 맞는 TC 프리미티브 객체(52)인 TC 시작 객체(52-1)를 실행시키게 된다(스텝 S64, S65)).

- <91> 이때, 해당 상위 TCAP 다이얼로그 상태 객체(34)는 휴지 상태에서 TC 시작 객체(52-1)를 실행하면서 SCP 관계 객체(33)를 통해 TC 시작 객체(52-1)의 어드레스 정보를 이용하여 해당되는 SCP를 찾은 후(스텝 S66), TCAP 다이얼로그를 전이중 상태로 천이시키게 된다(스텝 S67).
- <92> 한편, 해당 TCAP 메시지에 TC 컴포넌트도 존재하는 경우 해당 TCAP 메시지 객체(32)는 컴포넌트 실행 함수를 호출하여 나머지 TC 컴포넌트를 실행시키게 되며(스텝 S68), 해당 TC 프리미티브와 TC 컴포넌트를 실행시키면서 현재의 TCAP 다이얼로그 상태를 확인하게 되는데, 만약, 현재 TCAP 다이얼로그 상태가 종료 상태인 경우에는 다이얼로그 종료 함수를 실행시켜 TCAP 다이얼로그 객체(31)를 종료시킴으로써, SCP와의 다이얼로그를 종료시키게 된다(스텝 S69).
- <93> 나아가, 해당 TCAP 메시지의 TC 컴포넌트 실행 절차는 첨부한 도면 도 14와 같이 진행되는데, 해당 TC 컴포넌트를 실행시켜 INAP 오퍼레이션을 해당되는 어플리케이션에 적용시키게 된다.
- <94> 이를 위해 해당 TCAP 메시지 객체(32)는 컴포넌트 실행 함수를 차례대로 실행시키면서, 해당 TCAP 메시지에 존재하는 TC 컴포넌트를 실행시키게 되는데(스텝 S71), 이때, TC 컴포넌트 객체(41)는 TC 인보크 객체(46)가 상속받고, 해당 TC 인보크 객체(46)는 커넥션 관련 INAP 객체(46-1)가 상속받게 된다.
- <95> 즉, 해당 TC 컴포넌트를 실행시키는 것은 커넥션 관련 INAP 객체(46-1)를 실행시키는 것을 의미하는 것으로, 해당 INAP 객체(46-1)는 SCP 관계 객체(33)에게 상태 관련 메시지를 실행시키게 된다(스텝 S72).

- <96> 이에, 해당 SCP 관계 객체(33)는 자신이 가지고 있는 호 처리 객체에게 커백션 관련 INAP 객체(46-1)의 TC 인보크 이벤트가 발생함을 보고하게 되고(스텝 S73), 해당 호 처리 객체는 보고된 이벤트에 대해 자신이 가지고 있는 현재 호 처리 상태를 TC 인보크 관련 INAP 객체(46-1)로 리턴시키게 된다(스텝 S74).
- <97> 이때, 해당 TC 인보크 관련 INAP 객체(46-1)는 현재 호 처리 상태에 대한 TC 인보크 이벤트에 따라 INAP 커백션 관련 함수를 실행시켜 해당되는 호 처리를 수행하게 되는데, 해당 커백션에 대한 호 처리는 SSP의 호 처리 객체에 의해 수행된다(스텝 S75).
- <98> 한편, 상술한 스텝 S69에서 해당 SCP와의 다이얼로그를 종료시키는 절차는 첨부한 도면 도 15와 같이 진행되며, 이때 종료되는 다이얼로그와 관련된 TCAP 다이얼로그 객체(31) 및 SCP 관계 객체(33)를 제거하게 된다.
- <99> 해당 TCAP 다이얼로그 객체(31)가 상위 TCAP 다이얼로그 상태 객체(34)를 통해 다이얼로그 종료 함수를 호출하게 되면(스텝 S81, S82), 해당 다이얼로그 종료 함수는 SCP 관계 객체(33)에 호출되어 SCP 관계 종료를 나타내는 호 처리 이벤트를 자신이 상속하는 호 처리 객체에 생성하게 된다(스텝 S83).
- <100> 그리고, 해당 SCP 관계 객체(33)가 생성된 SCP 관계 종료 이벤트에 호 진행 함수를 실행시키게 되면(스텝 S84), 해당 SCP 관계 종료 이벤트는 호 처리 객체에 이벤트 발생을 보고하게 되며(스텝 S85), 해당 호 처리 객체는 보고된 이벤트에 대해 자신이 가지고 있는 현재 호 처리 상태를 SCP 관계 종료 이벤트로 리턴시키게 된다(스텝 S86).
- <101> 이에, 해당 SCP 관계 종료 이벤트가 현재 호 처리 상태에 SCP 관계 종료 함수를 호출하게 되면(스텝 S87), 해당 호 처리 상태는 SCP 관계 제거 함수를 호출하여 해당 호

처리 객체가 가지고 있는 SCP 관계 객체(33)를 제거하게 된다(스텝 S88).

<102> 이때, 해당 SCP 관계 종료 이벤트의 역할이 종료됨에 따라 삭제 오퍼레이션을 이용하여 SCP 관계 종료 이벤트를 삭제한 후(스텝 S89), TCAP 다이얼로그 관리 객체(61)를 관리하고 있던 SCP 관계 객체(33)를 삭제하게 된다(스텝 S90).

<103> 한편, 상술한 SSP의 INAP 소프트웨어 블록에 의한 TC 시작을 나타내는 TC 프리미티브 및 TC 인보크를 나타내는 TC 컴포넌트의 처리 절차와 동일한 방법으로, 그 이외의 TC 프리미티브 및 TC 컴포넌트를 처리할 수 있는데, 이러한 메시지 처리는 항상 TC 프리미티브를 먼저 처리하면서 TCAP 다이얼로그 상태를 천이 또는 유지하고, 그 이후에 TC 컴포넌트를 처리하게 된다.

<104> 또한, 본 발명에 따른 실시예는 상술한 것으로 한정되지 않고, 본 발명과 관련하여 통상의 지식을 가진자에게 자명한 범위내에서 여러 가지의 대안, 수정 및 변경하여 실시할 수 있다.

【발명의 효과】

<105> 이상과 같이, 본 발명은 SSP에서 INAP을 이용하는 각종 어플리케이션들이 INAP 메시지를 이용하여 TCAP와 자유롭게 통신할 수 있게 되며, 해당 INAP 처리를 위한 소프트웨어 블록의 객체지향적인 설계, 구현을 통해 재사용성을 증대시킴으로써, 이를 이용하는 어플리케이션의 변경없이 INAP 메시지만을 추가, 변경, 삭제하여 새로운 시스템에 적용할 수 있게 됨과 동시에 유지 보수가 간편해진다.

【특허청구범위】**【청구항 1】**

INAP 팩터리 객체를 통해 INAP 초기 감지점과 관련된 INAP 객체를 생성하는 과정과; 상기 INAP 객체를 파라미터로 송신 컴포넌트 함수를 호출하여 인보크 ID와 다이얼로그 ID를 TCAP 다이얼로그 객체로부터 얻어 상기 INAP 메시지에 셋팅하는 과정과; 컴포넌트 추가 명령을 내려 TCAP 메시지 객체에 상기 INAP 객체를 추가하는 과정과; 현재의 TCAP 다이얼로그 상태에 따라 서로 다른 송신 TCAP 이벤트를 생성 및 실행시켜 상기 INAP 메시지를 송신한 후, 삭제하는 과정을 포함하는 것을 특징으로 하는 에스에스피에서 티씨에이피와 통신하기 위한 아이엔에이피 처리 방법.

【청구항 2】

제 1항에 있어서,

상기 송신 TCAP 이벤트를 생성 및 실행시켜 상기 INAP 메시지를 송신한 후, 삭제하는 과정은, TCAP 다이얼로그 휴지 상태에 송신 TCAP 이벤트를 생성하는 단계와; TC 컴포넌트 객체로부터 INAP 객체를 상속받아 TC 시작을 나타내는 TC 프리미티브 객체를 생성하는 단계와; SCP 관계 객체로부터 TC 시작에 필요한 정보를 얻어 TC 프리미티브에 셋팅하는 단계와; TC 프리미티브 추가 명령을 내려 이미 TC 컴포넌트가 추가된 TCAP 메시지에 TC 프리미티브를 추가하여 송신하는 단계와; 초기 상태 천이 명령을 내려 상기 TCAP 다이얼로그 객체를 휴지 상태에서 초기 상태로 천이하는 단계를 더 포함하는 것을 특징

으로 하는 에스에스피에서 티씨에이피와 통신하기 위한 아이엔에이피 처리 방법.

【청구항 3】

제 2항에 있어서,

상기 TC 컴포넌트가 추가된 TCAP 메시지에 TC 프리미티브를 추가하여 송신하는 단계는, 상기 TC 컴포넌트를 TCAP 및 그 하부 네트워크 프로토콜에서 사용할 수 있는 포맷으로 가공하는 단계와; 인보크 요구 메시지의 데이터 영역에 가공된 TC 컴포넌트를 실어 TCAP으로 송신하는 단계와; 상기 TCAP으로 모든 TC 컴포넌트를 송신한 후, 상기 TCAP 메시지의 TC 프리미티브를 송신하는 단계를 더 포함하는 것을 특징으로 하는 에스에스피에서 티씨에이피와 통신하기 위한 아이엔에이피 처리 방법.

【청구항 4】

제 3항에 있어서,

상기 TCAP 메시지의 TC 프리미티브를 송신하는 단계는, 인코딩 함수를 호출하여 TC 프리미티브 시작 상태로 천이하는 단계와; 상기 TC 프리미티브를 TCAP 및 그 하부 네트워크 프로토콜에서 사용할 수 있는 포맷으로 가공하는 단계와; 송신 시작 요구 메시지의 데이터 영역에 가공된 TC 프리미티브를 실어 TCAP으로 송신하는 단계를 더 포함하는 것을 특징으로 하는 에스에스피에서 티씨에이피와 통신하기 위한 아이엔에이피 처리 방법.

【청구항 5】

TCAP으로부터 TC 프리미티브가 수신되는 경우 수신된 TC 프리미티브를 처리하기 위한 TC 시작 객체를 생성하는 과정과; 상기 TC 시작 객체를 실행시켜 수신된 TC 프리미티브에 포함된 다이얼로그 ID를 이용하여 해당되는 SCP 관계 객체를 찾는 과정과; 상기 SCP 관계 객체로 TC 시작 추가를 명령하여 TCAP 메시지에 TC 시작 객체를 추가시키는 과정과; TC 컴포넌트없이 TC 프리미티브만 수신된 경우 상기 TC 시작 객체가 추가된 TC 프리미티브를 실행시키는 과정과; 해당되는 호 처리를 수행한 후, 상기 SCP 관계 객체를 삭제하여 SCP와의 다이얼로그를 종료시키는 과정을 포함하는 것을 특징으로 하는 에스에스피에서 티씨에이피와 통신하기 위한 아이엔에이피 처리 방법.

【청구항 6】

제 5항에 있어서,

상기 TC 시작 객체를 생성하는 과정은, 수신된 TC 프리미티브를 디코딩하여 얻은 TC 시작 객체와 동일한 TC 시작 객체를 생성하는 것을 특징으로 하는 에스에스피에서 티씨에이피와 통신하기 위한 아이엔에이피 처리 방법.

【청구항 7】

제 5항에 있어서,

상기 수신된 TC 프리미티브에 포함된 다이얼로그 ID를 이용하여 해당되는 SCP 관계 객체를 찾는 과정에서, 상기 다이얼로그 ID를 찾지 못한 경우 TCAP 다이얼로그 객체를

통해 새로운 SCP 관계 객체를 생성하는 과정을 더 포함하는 것을 특징으로 하는 에스에스피에서 티씨에이피와 통신하기 위한 아이엔에이피 처리 방법.

【청구항 8】

제 5항에 있어서,

상기 TC 시작 객체가 추가된 TC 프리미티브를 실행시키는 과정에서, TC 프리미티브와 함께 TC 컴포넌트가 수신된 경우 상기 TC 프리미티브의 실행을 보류한 상태에서 모든 TC 컴포넌트를 처리한 후, 보류되었던 TC 프리미티브를 실행시키는 과정과; TC 프리미티브의 실행을 보류한 상태에서 처리한 TC 컴포넌트를 실행시키는 과정을 더 포함하는 것을 특징으로 하는 에스에스피에서 티씨에이피와 통신하기 위한 아이엔에이피 처리 방법.

【청구항 9】

제 8항에 있어서,

상기 TC 컴포넌트를 처리하는 과정은, TCAP으로부터 TC 컴포넌트가 수신되는 경우 수신된 TC 프리미티브에 포함된 TC 인보크 객체에 실려있는 오퍼레이션 코드를 분석하는 단계와; INAP 커넥션 관련 오퍼레이션 코드인 경우 커넥션 관련 INAP 객체를 생성하는 단계와; 상기 INAP 객체를 실행시켜 수신된 TC 컴포넌트에 포함된 다이얼로그 ID를 이용하여 해당되는 SCP 관계 객체를 찾는 단계와; 상기 SCP 관계 객체로 TC 컴포넌트 추가를 명령하여 TCAP 메시지에 TC 컴포넌트 객체를 추가시키는 단계와; 마지막 TC 컴포넌트인

경우 이를 실행시키는 더 단계를 포함하는 것을 특징으로 하는 에스에스피에서 티씨에 이피와 통신하기 위한 아이엔에이피 처리 방법.

【청구항 10】

제 5항 또는 8항에 있어서,

상기 TC 프리미티브를 실행시키는 과정은, TC 시작을 나타내는 TC 프리미티브 이벤트를 발생시키고, 휴지 상태인 TCAP 다이얼로그 상태를 리턴받는 단계와; TC 시작 객체의 어드레스 정보를 이용하여 해당되는 SCP를 찾은 후, 상기 TCAP 다이얼로그 상태를 전이중 상태로 천이시키는 단계와; TCAP 메시지에 TC 컴포넌트가 존재하는 경우 해당 TC 컴포넌트를 실행시키는 단계와; 상기 TCAP 다이얼로그 상태가 종료 상태로 천이되는 경우 TCAP 다이얼로그 객체를 종료시켜 SCP와의 다이얼로그를 종료시키는 단계를 더 포함하는 것을 특징으로 하는 에스에스피에서 티씨에이피와 통신하기 위한 아이엔에이피 처리 방법.

【청구항 11】

제 8항에 있어서,

상기 TC 컴포넌트를 실행시키는 과정은, 컴포넌트 실행 함수를 실행시켜 TC

컴포넌트객체 및 커넥션 관련 INAP 객체를 상속하는 단계와; SCP 관계 객체를 통해 TC 인보크 이벤트 발생을 보고하는 단계와; 보고된 이벤트에 대해 현재 호 처리 상태를 TC 인보크 관련 INAP 객체로 리턴시키는 단계와; 현재 호 처리 상태에 대한 TC 인보크 이벤트에 따라 해당되는 호 처리를 수행하는 단계를 더 포함하는 것을 특징으로 하는 에스에스피에서 티씨에이피와 통신하기 위한 아이엔에이피 처리 방법.

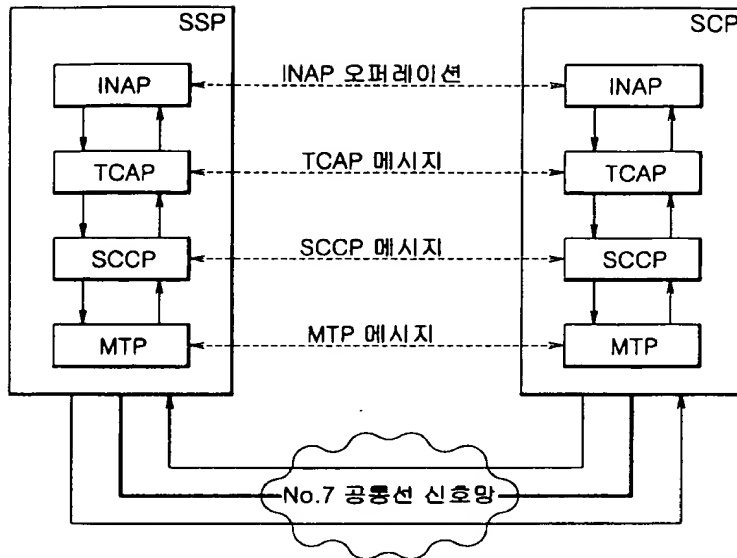
【청구항 12】

제 5항에 있어서,

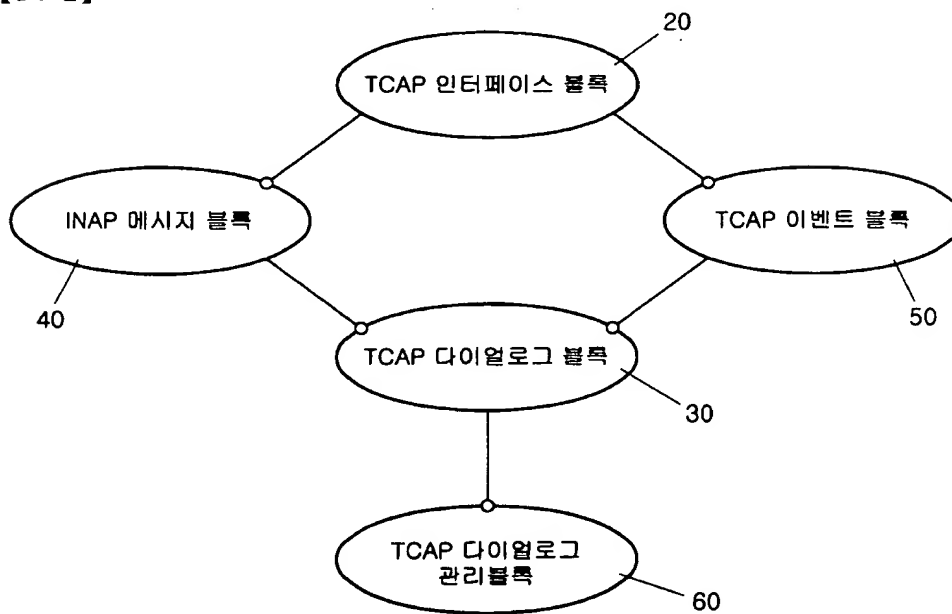
상기 SCP와의 다이얼로그를 종료시키는 과정은, 다이얼로그 종료 함수를 호출하여 SCP 관계 종료를 나타내는 호 처리 이벤트를 생성하는 단계와; 생성된 SCP 관계 종료 이벤트를 호 처리 객체에 보고하여 현재 호 처리 상태를 리턴받는 단계와; 상기 SCP 관계 종료 이벤트를 삭제한 후, 해당되는 SCP 관계 객체를 삭제하여 SCP와의 다이얼로그를 종료시키는 단계를 더 포함하는 것을 특징으로 하는 에스에스피에서 티씨에이피와 통신하기 위한 아이엔에이피 처리 방법.

【도면】

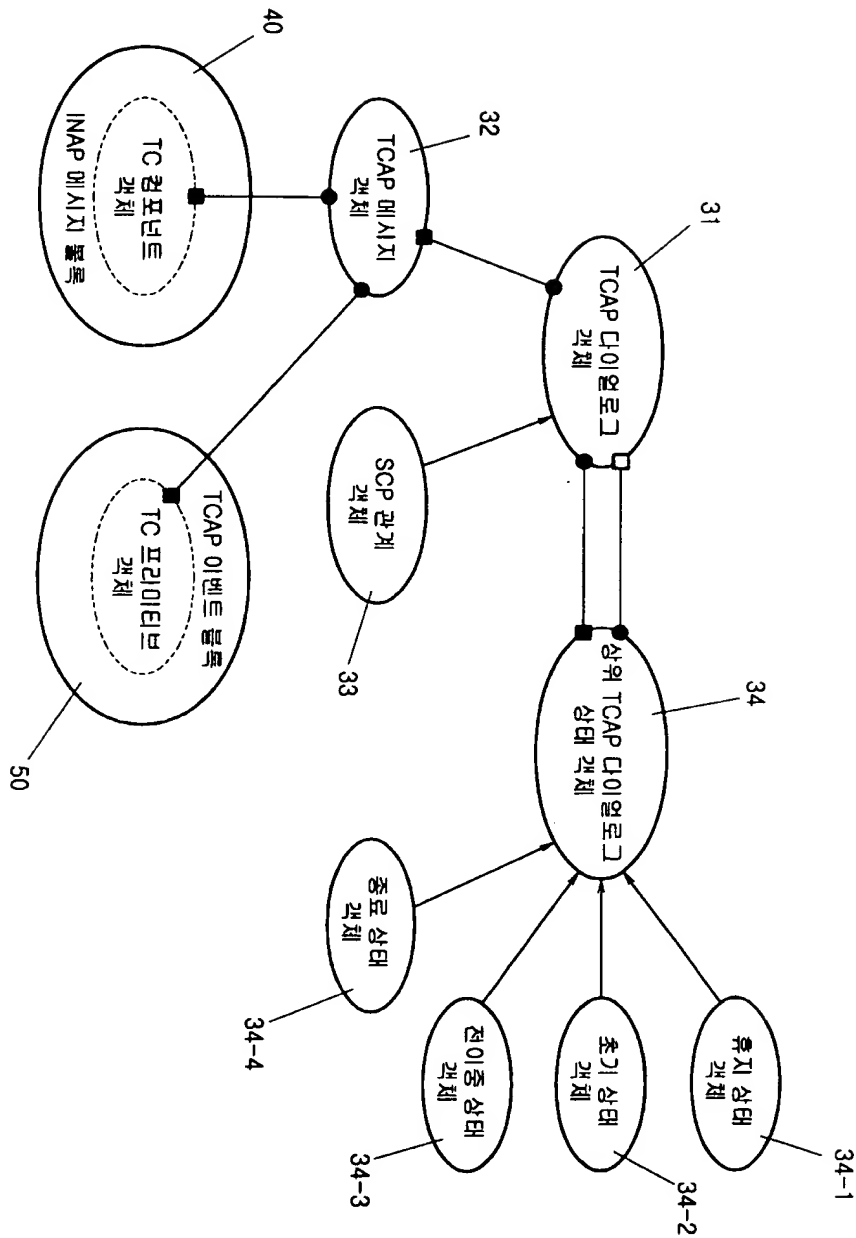
【도 1】



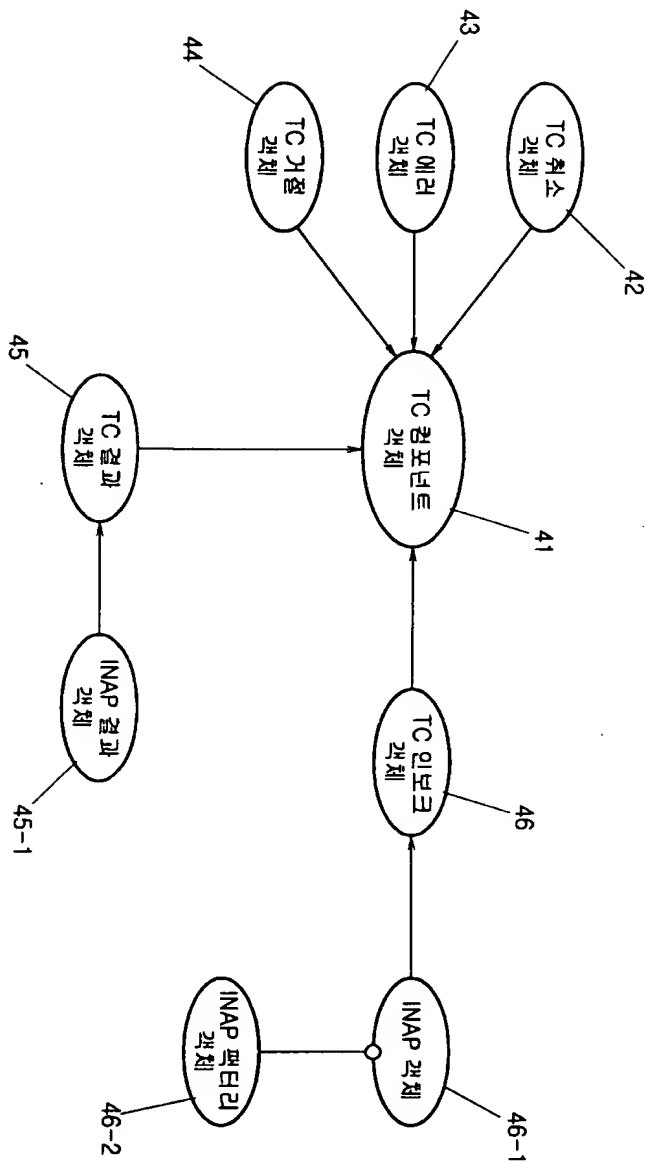
【도 2】



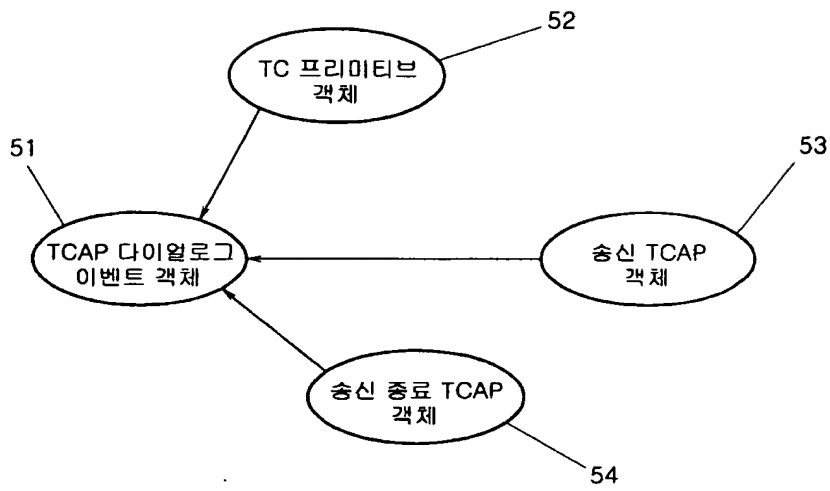
【도 3】



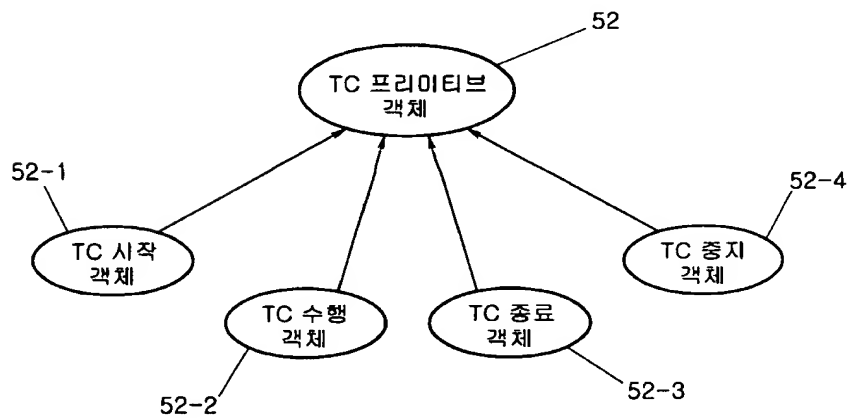
【图 4】



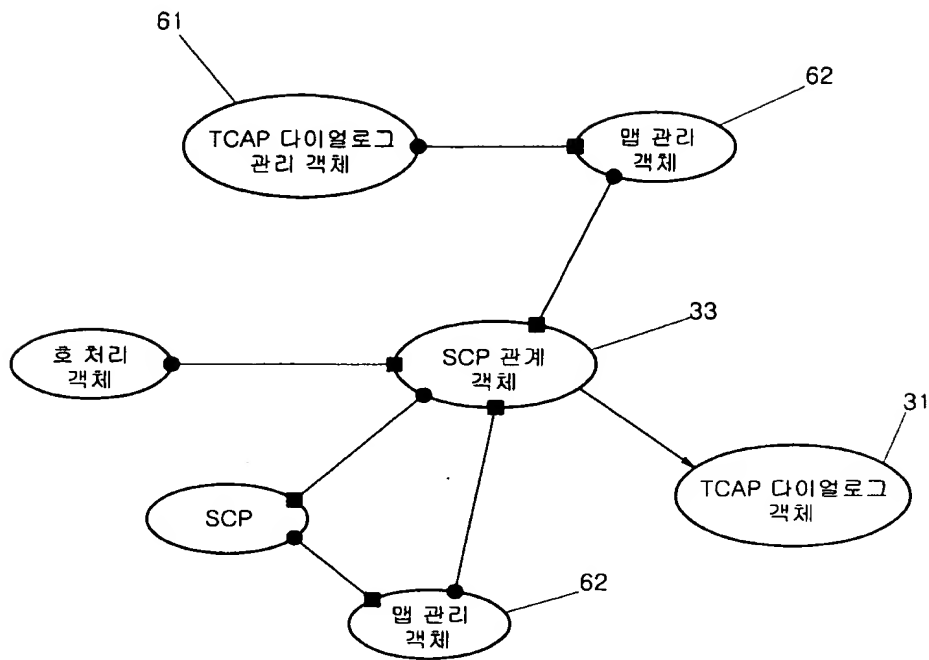
【도 5】



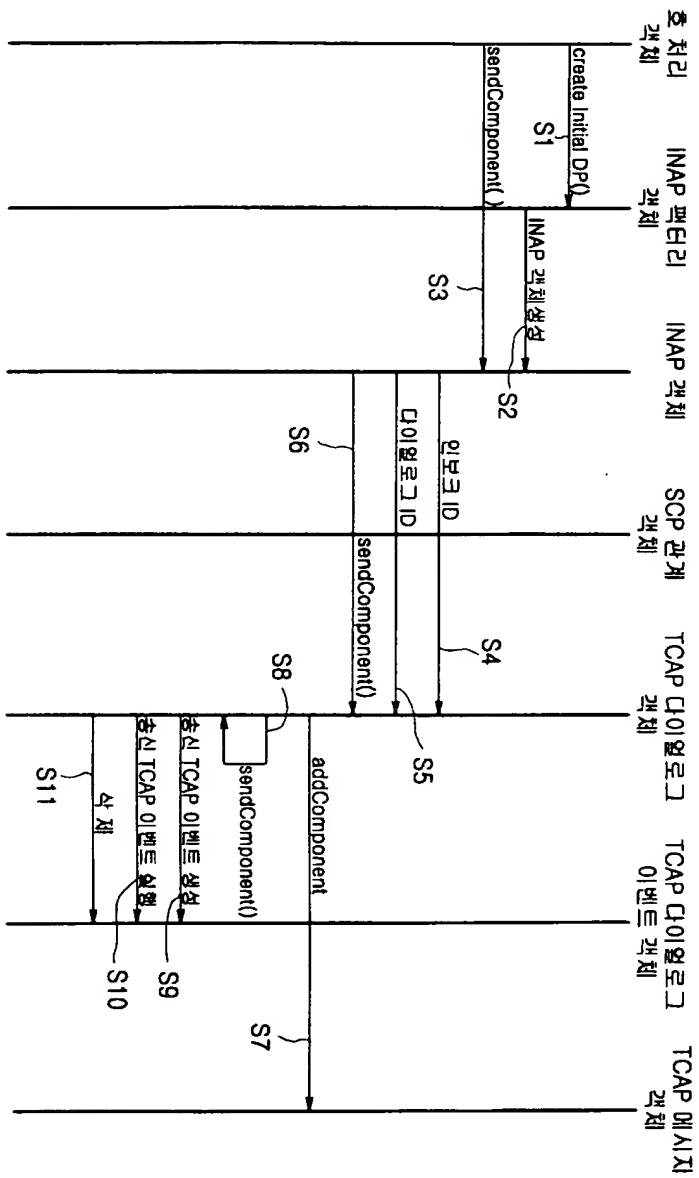
【도 6】



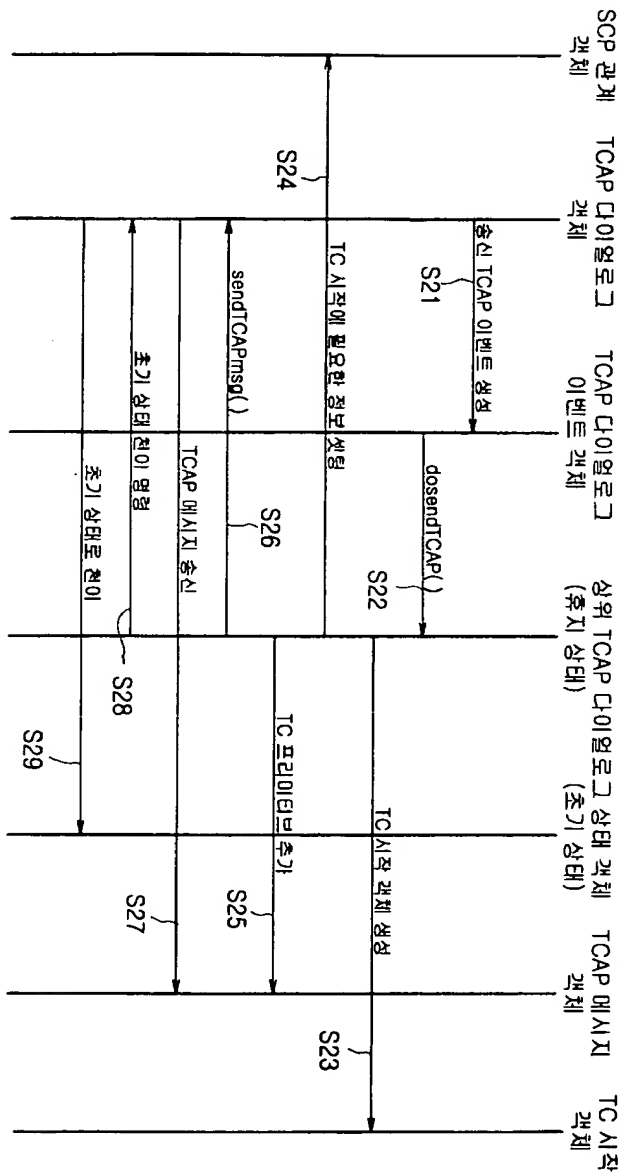
【도 7】



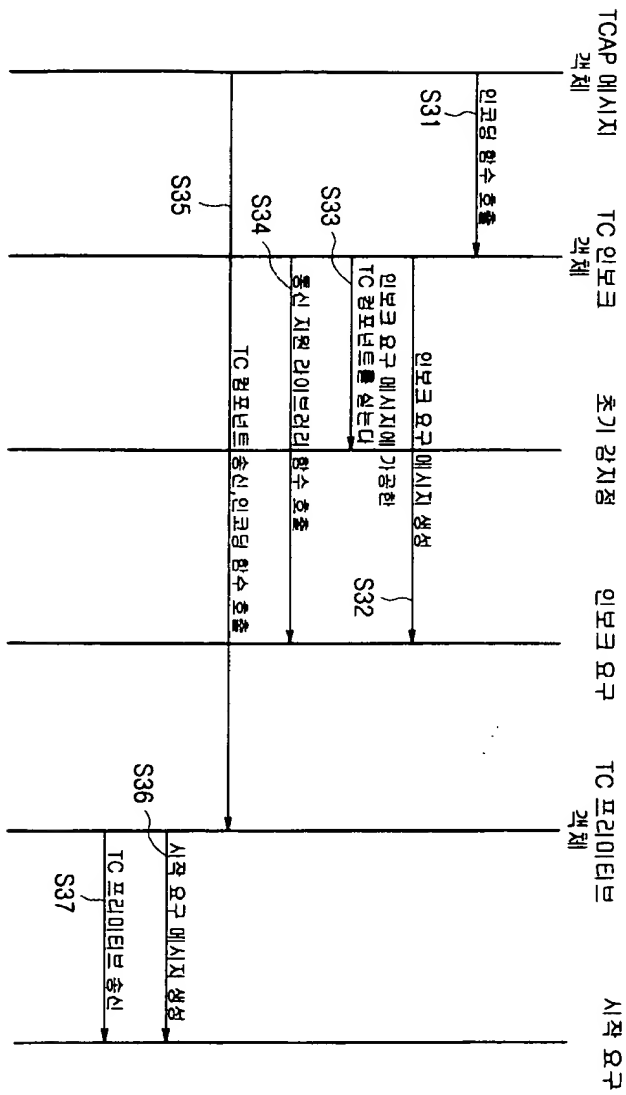
【도 8】



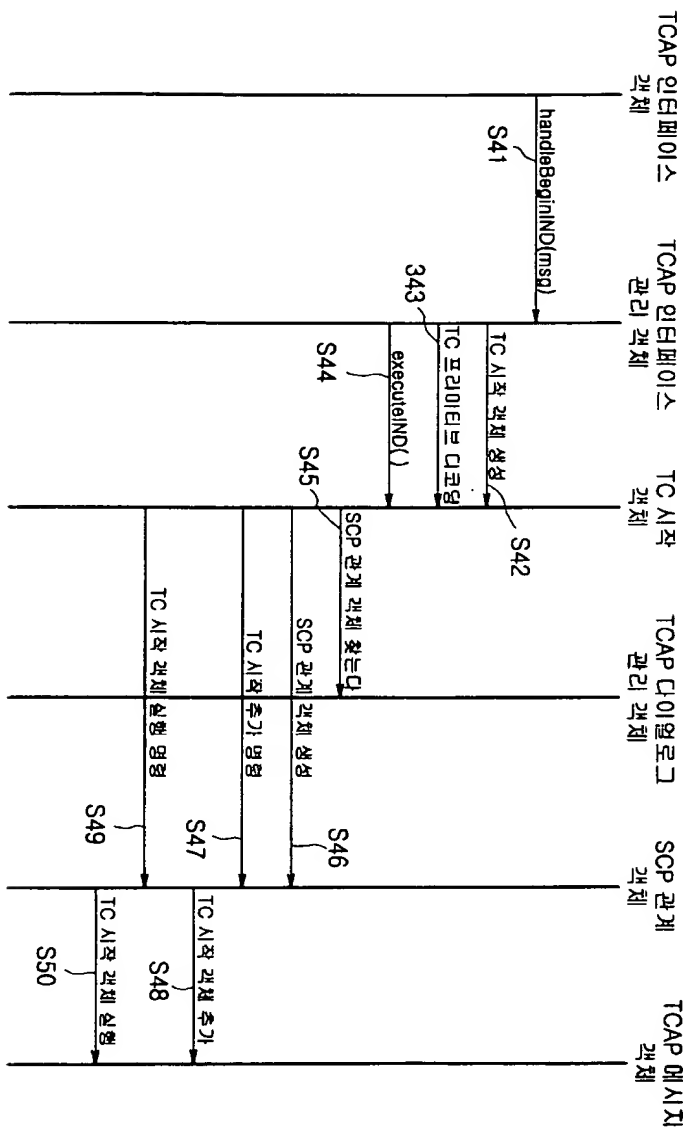
【도 9】



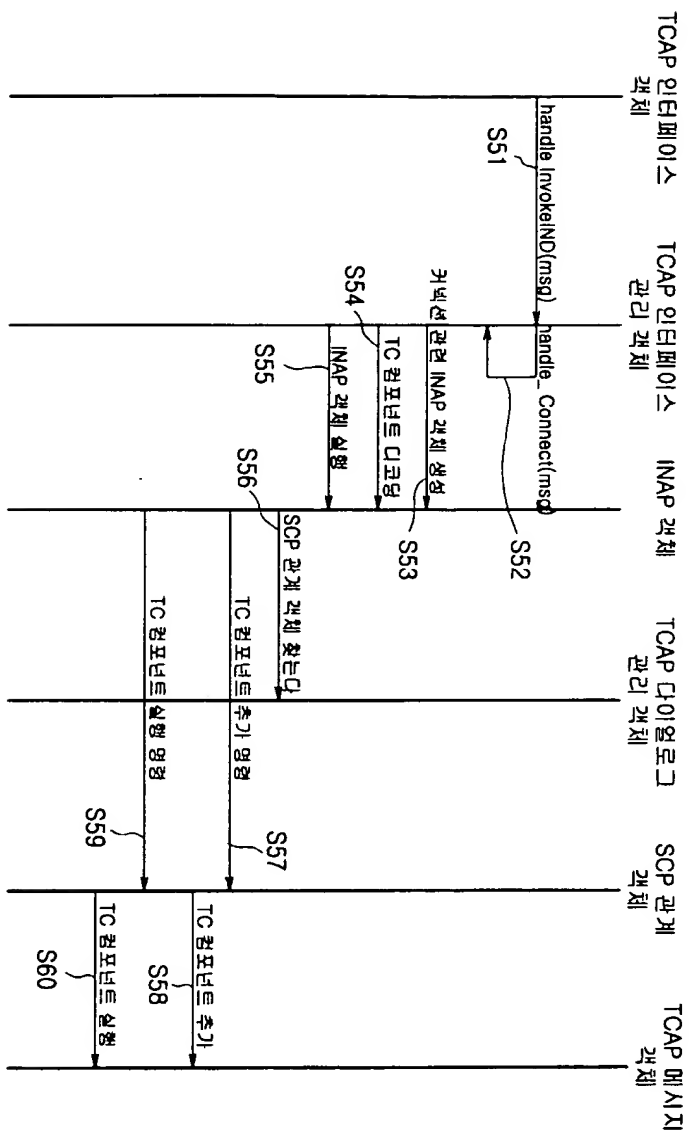
【도 10】



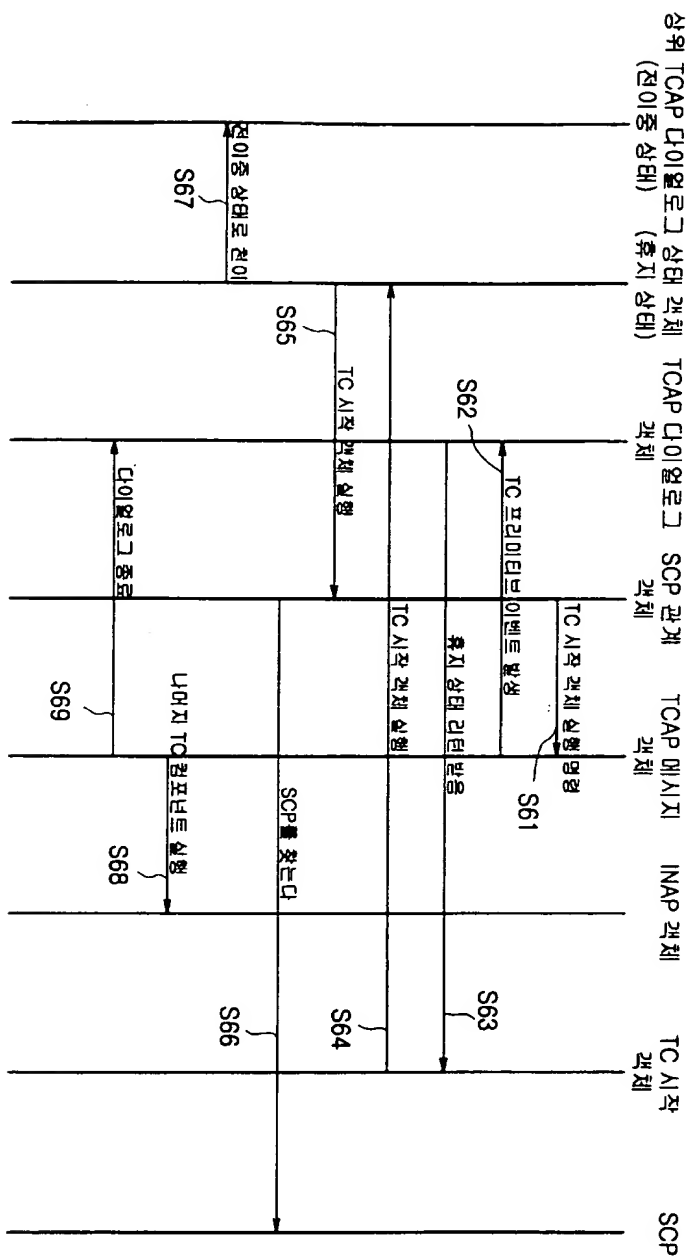
【도 11】



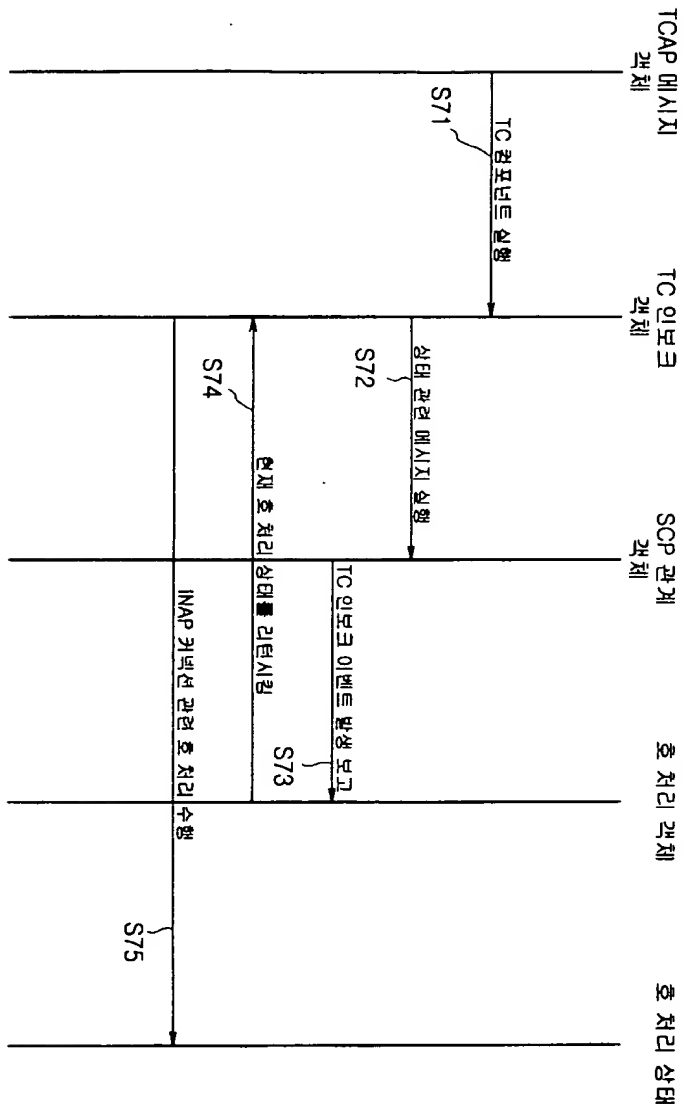
【도 12】



【도 13】



【도 14】



【도 15】

